4

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AIM 1050	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) What are plans for?		5. TYPE OF REPORT & PERIOD COVERED memorandum
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Philip E. Agre and David Chapman		8. CONTRACT OR GRANT NUMBER(s) N00014-85-K-0124
9. PERFORMING ORGANIZATION NAME AND ADDRESS Artificial Intelligence Laboratory 545 Technology Square Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209		12. REPORT DATE September 1988
		13. NUMBER OF PAGES 18
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Arlington, VA 22217		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Unlimited

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

planning
improvisation
action
instruction following

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

What plans are like depends on how they are used. We contrast two views of plan use. On the plan-as-program view, plan use is the execution of an effective procedure. On the plan-as-communication view, plan use is like following natural language instructions. We have begun work on computational models of plans-as-communication, builiding on our previous work on improvised activity and on ideas from sociology/

Massachusetts Institute of Technology
Artificial Intelligence Laboratory

A.I. Memo 1050 September 1988

Accession For		
NTIS GRA&I	X	
DTIC TAB	☐	
Unannounced	☐	
Justification		
By		
Distribution/		
Availability Codes		
Dist	Avail and/or Special	
A-1		

What are plans for?

Philip E. Agre
David Chapman

Abstract

What plans are like depends on how they're used. We contrast two views of plan use. On the plan-as-program view, plan use is the execution of an effective procedure. On the plan-as-communication view, plan use is like following natural language instructions. We have begun work on computational models of plans-as-communication, building on our previous work on improvised activity and on ideas from sociology.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract number N00014-85-K-0124.

1 Introduction

What plans are like depends on how they're used. ~~We contrast~~ two views of plan use are contrasted. On the plan-as-program view, plan use is the execution of an effective procedure. On the plan-as-communication view, plan use is like following natural language instructions. We have begun work on computational models of plans-as-communication, building on our previous work on improvised activity and on ideas from sociology.

The plan-as-program view and the plan-as-communication view offer very different accounts of the role of plans in activity. The plan-as-program view gives plans a central role. Plan use is only a matter of *execution*, performed by a simple, fixed, domain-independent "interpreter." Plans-as-programs directly determine their user's actions.

The plan-as-communication view gives plans a much smaller role. It requires an account of *improvisation*. Plans, on this account, do not directly determine their user's activity. Indeed, an agent can engage in sensible, organized, goal-directed activity without using plans at all. An agent who *does* use a plan-as-communication does not mechanically execute it. Instead, the agent uses the plan as one resource among others in continually redeciding what to do. Using a plan requires figuring out how to make it relevant to the situation at hand, a process of interpretation which can be arbitrarily complex.

Section 2 of this essay describes the plan-as-program view and some of the difficulties with it. These difficulties concern the computational complexity of plan construction, the problem of prediction in a world of uncertainty and change, the necessity of accommodating the stupidity of executives by specifying plans in impractical detail, and the largely unaddressed issue of relating plan texts to concrete situations in the world.

Section 3 outlines our view that everyday activity is fundamentally improvised. Improvisation might involve ideas about the future, but in any event it requires a continual redecision about what to do *now*. Supporting this process of continual redecision is a technical problem that we have addressed in our work. We briefly describe our most recent project, a program called Pengi that employs novel kinds of perception and representation in playing a commercial video game called Pengo.

Section 4 presents the plan-as-communication view and contrasts its views of plan use, representation, and activity with those of the plan-as-program view. It further illustrates the view with an example of plan use in the real world. Our analysis of this example turns up many ways in which the plan's maker counted on the understandings of everyday reality that he shared with the plan's users.

Section 5 pursues this theme in more detail in relation to our current work. Building on our work on Pengo-playing, we have been trying to understand the role of situated language use in the activity of playing cooperative video games. We describe some of what we've observed in watching real players and offer the beginnings of what we hope will become a more systematic account of the shared reality of plan makers and plan

users.

Section 6 summarizes our principal conclusions and proposes that future inquiry conjoin computational analysis and model-building with principled and detailed observation of cooperative situated language use in natural settings.

This paper is not intended as a thorough survey of the literature on planning. For useful surveys see (Chapman 1987), (Tate 1985), and (Swartout 1988).

2 Plans as programs

The plan-as-program view understands plan use as program execution. Almost all implemented executives have been modeled on programming language interpreters. A plan language, on this view, is like a programming language. The plans are built out of a set of parameterized primitives (such as PUT–ON(x,y)) using a set of composition operators (to indicate serial execution, for example). Executing a plan means walking over it in a "syntactic," "mechanical" fashion, performing its primitive actions and monitoring conditions specified by the planner. The executive is domain-independent: it applies no domain knowledge except that implicit in the plan. It makes no interpretations of its sensor inputs except for the monitored conditions and any predicates that might appear in plan conditionals. Nor does it second-guess the planner by performing any interpolations, substitutions, or rearrangements that would count as a departure from the plan. If the executive gets into trouble, it gives up and returns control to the planner. In short, the planner is smart and the executive is dumb.

(The plan-as-program view implies domain-independent *plan execution*, not domain-independent *plan construction*. Plan-as-program construction can be domain-independent or domain-dependent, algorithmic or case-based, formally correct or heuristic.)

This section discusses four reasons to doubt the plan-as-program view. (1) The view poses computationally intractable problems. (2) It is inadequate for a world characterized by unpredictable events such as the actions of other agents. (3) It requires that plans be too detailed. (4) Finally, it doesn't address the problem of relating the plan text to the concrete situation.

(1) The plan-as-program view makes planning into automatic programming with all its formal undecidabilities. Chapman (1987) has proven some negative complexity results, both about the manipulations that need to be performed on partially specified plans and about the spaces through which plan-as-program planners must search. As formalizations of actions and preconditions become more realistic, these results get worse quickly.

(2) The original planners made plans to achieve goals in very well-behaved simulated worlds. In these imaginary worlds, it was possible to construct a plan which consisted

of a representation of a sequence of primitive actions, which, performed in order, would provably achieve the goal. Thus it was possible to formulate the "planning problem" in terms of constructing something that would, when "executed," "control" the robot. (For a review of this literature see Chapman 1987.) It has been widely recognized in the last few years that in the real world blind execution is impossible because unpredictable external processes can change the world and causally affect the robot.

Plans-as-programs are not very flexible. If the robot's interactions with its world don't work out as the planner expected, the plan won't work. If the planner explicitly anticipates a specific, detectable uncertainty, it can provide the plan with a conditional branch. If it doesn't, then a new plan will be required. Reasons to abort or revise a plan can be divided into two classes, contingencies and opportunities. If you're about to walk through the kitchen door to fetch a pen, a closed door is a contingency and a pen on a desk just outside the kitchen is an opportunity. Not all contingencies can be detected through precondition failures: maybe you can put your pants on over your shoes without violating any preconditions, but it's usually not sensible. Opportunities are still harder to test for because they're less obtrusive. An enormous range of circumstances might count as opportunities in one situation or another. In short, a new plan is called for whenever it isn't sensible to continue following the existing one. This is a grave problem for any executive that isn't as smart and knowledgeable as its planner.

(3) It is generally acknowledged that no system could produce completely detailed plans in domains of realistic complexity. Real activity is too complicated for that. It follows that an executive has to be expected to fill in some *details as it goes along*. It also follows that a planner needs some idea of what details it can rely on its executive to fill in. A dumb executive will need everything spelled out for it, but if the executive were smarter the planner could paint the desired actions with a broader brush. Ideally, a planner would only have to deal with issues that the executive can't. Its plans would not be laden with redundant details. Nor would they prejudge decisions better left to the executive, which after all can base its judgements on the world as it actually turns out, not on models of projected worlds.

The plan-as-program view offers us one account of how a plan can be operational without spelling out every detail. If plans are like programs, then we can make compact plans using a hierarchy of subplans. The planner has a library of subplans, each of which has a contract. These contracts establish a partition between the issues that must concern the planner and the issues that subplans can deal with themselves. They enable the planner to live in a simple, abstract world, reasoning with the preconditions and effects of the top-level subplans.

This subplan-hierarchy view of plans has a number of shortcomings. First, the executive still cannot depart from the plan other than to return control to the planner. Second, it is unclear what sorts of domains permit hierarchical abstraction. The library

subplans have to satisfy their contracts, regardless of the specific circumstances; this makes them very difficult to construct. In complex real-world domains, where enormous numbers of concrete contingencies can bear on abstract goal ordering issues, truly hierarchical decomposition may not be possible. (See Lozano-Pérez and Brooks 1985 for further discussion.)

(4) An executive has to establish a causal connection between the text of the plan it is executing and the materials in the concrete situation in front of it. The ontologies of most existing plan languages posit a world made up of individuals, some of which correspond to constant symbols in the agent's axiom set. Thus, for example, the truth of a typical blocks-world proposition like ON(A,B) is determined by a relation corresponding to ON applied to individuals corresponding to A and B. A plan might achieve the goal ON(A,B) by executing an action like PUT–ON(A,B). This requires that the executive be able to determine automatically which individuals in its world correspond to the constant symbols A and B. If every object has a bar-code affixed to it then it's easy enough. But blocks on tables and luggage in airports and cars in parking lots and turns on highways very often take work to distinguish. Arbitrary domain knowledge can, and regularly does, enter into determining which object is the one you want.

Not only does the practice of allowing primitive actions to traffic in constant symbols beg this problem, it masks a still deeper one. Much of the work of using a plan is in determining its *relevance* to the successive concrete situations that occur during the activity it helps to organize. By hiding this work, an executive that can automatically relate symbols to objects radically falsifies the nature of plan use. Plan use requires domain-specific skills that a programming language interpreter does not possess and situation-specific improvisations that a programming language interpreter cannot perform.

With many AI researchers trying to work through their dissatisfactions with traditional formulations of planning, recent research has developed the notions of "interleaved" or "incremental" planning (Chien and Weissman 1975, Giralt *et al* 1984, McDermott 1978, Tate 1984, Wilensky 1983, Wilkins 1985 and 1988) and of "reactive" or "tactical" planning (Firby 1987, Fox and Smith 1984, Georgeff and Lansky 1986 and 1987). (More generally, the word "planning" is being stretched in many different directions at once. Many people, for example, seem to be using "planning" to mean "sensibly acting." We feel that the word "planning" ought, at a minimum, to imply that a plan is involved. We are also unhappy with the phrase "reactive planning," which is a contradiction in terms.)

In interleaved planning the planner makes its plan as always. When the executive gets into trouble, it passes control back to the planner, which assesses the situation and makes a new plan. Interleaved planning continues the theme of control. The plan specifies an ideal future history; the executive's job is to try to force the world to conform to it as much as possible. The executive must defensively monitor the situation to detect whether

things have gone wrong. The outside world is something that gets in the way, that makes control difficult, that frustrates the robot's carefully laid plans and sends it back to the drawing board. An interleaved planner is reactive, in a pejorative sense: it does work only in breakdown rather than creatively making use of opportunities and contingencies. Interleaved planning is like waiting for your car to hit something before bothering to change direction.

"Reactive" and "tactical" are new terms whose meanings are not yet clear. Many of these systems are not planners in any useful sense; some do not try to anticipate the future at all. In fact, most resemble the executive part of an interleaved planning system together with an externally generated plan library.

We suspect that much of the appeal of the plan-as-program view originates in the word "execution." To execute a command or instruction is to carry it into effect; to execute an action or operation is to perform it. The word is little used except in legal and administrative senses, but even its broader use suggests an activity that takes place in a narrowly specified institutional context, with articulated constraints and strict criteria, and with negligible room and need for variation, interpretation, improvisation, or any other deviation on the part of the person doing the executing. To "execute" a plan isn't just to "follow" it, it's to follow it "to the letter" and "by the book."

The principal source of the word "execution" in AI research is Miller, Galanter, and Pribram's extraordinarily influential book *Plans and the Structure of Behavior* (1960). The word "execution" has exerted a continual unspoken pressure on the attention of AI people, leading us to think of a plan as a pretty-well-thorough representation of a sequence of actions, so that execution is a simple process. It is very tempting to assimilate plan execution to running a program on an interpreter; perhaps the word "execution" successfully kept all but a few people from becoming dissatisfied with this idealization.

3 Participation

What might an alternative to plans-as-programs look like? Let's start by ditching the word "execution." It tends to prejudge issues by making the plan-as-program view seem inevitable. Instead, let's simply speak of people (or robots) "using" plans. This simple terminological change makes some hard questions seem more urgent. First, what can one do with a plan besides trying to mechanically execute it? Second, how do plans and plan-making change if plan users can be counted on to use plans sensibly rather than mechanically marching through them?

We don't know if these questions must have the same answers for robots as they do for people. But so long as alternatives to the plan-as-program view are in short supply, evidence from human plan use can bring some perspective. Most of what is known

has been discovered by social scientists such as Gladwin (1970), Hutchins (1987), Scher (1984), Suchman (1986, 1987), Scribner's group (Scribner 1984, Beach 1986), and the Soviet activity theorists (Wertsch 1985).

The plan-as-program view gives plans a central role in determining activity. In particular, it claims that an agent acts as it does because it has a certain plan. We do not believe this claim. According to the plan-as-communication view, a plan does not directly determine an agent's actions. Instead, a plan is a resource that an agent can use in deciding what to do. What, then, *does* determine an agent's actions? Answering this question is the job of a *theory of activity.* After briefly summarizing our understanding of activity in this section, we will return to the question of the role of plans in activity.

Our theory of activity has two interconstraining parts: a theory of cognitive machinery and a theory of the *dynamics* or regularly occurring patterns of activity. In studying people we ask (i) how is ordinary human activity organized and (ii) what does this imply for the organization of human cognitive machinery? In studying machines we ask (i) what forms might an agent's activity take and (ii) what sorts of cognitive machinery are compatible with what sorts of activity?

Our answers to these questions are informed by the central theme of participation in ongoing activity whose determination is shared with other processes and agents. Everyday routine activity, we believe, has an orderliness and coherence that is independent of any plan or other representation of it. See (Chapman and Agre 1986) for some of this story and (Agre forthcoming) for much of the rest. We've found that participating in the flow of the environment, rather than attempting to control it, can radically simplify the machinery required to account for the organization of activity.

We built the Pengi system (Agre and Chapman 1987, and forthcoming extended version) to illustrate some of what we've learned. Though Pengi engages in complex patterns of activity, its machinery is extremely simple: a visual system based on psychophysically motivated ideas from Ullman's visual routines theory (1982), a simple motor system, and a central system made entirely of combinational logic.

Pengi does not follow any plans, but neither is it pushed around by its world. The Pengo games Pengi plays move fast, so Pengi constantly uses the contingencies and opportunities of its environment to help it *improvise* ways to pursue its projects. Improvisation differs from planning-as-programming in that each moment's action results, effectively, from a fresh reasoning-through of that moment's situation. Yet improvisation, like planning, involves ideas about what might happen in the future.[1]

One of Pengi's contributions is a new participatory theory of representation called *indexical-functional*, or *deictic*, representation (Agre and Chapman 1988). Whereas traditional representations posit an ill-characterized "semantic" correspondence between

[1]The extreme abbreviation of our published papers has led to some confusion on this point. For example Firby (1987, p. 203) incorrectly ascribes to us the belief that "[c]omplex activity arises from the continual activation of actions with no anticipation of the future."

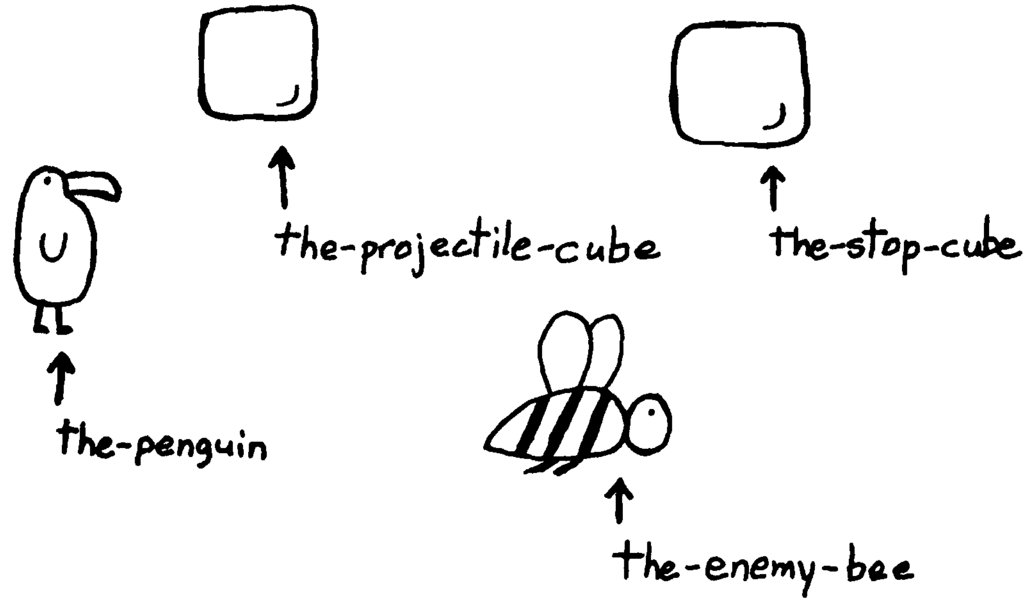

Figure 1. A Pengo situation that requires looking ahead.

symbols in an agent's head and objectively individuated objects in the world, our theory describes a causal relationship between the agent and indexically and functionally individuated *entities* in the world. For example, one of the entities Pengi recognizes is *the-bee-I-am-chasing*. This entity is individuated indexically in that it is defined in terms of its relationship to the agent ("I"). It is also individuated functionally in that it is defined in terms of one of the agent's ongoing projects (chasing a bee). Whereas in a traditional representation, the symbols BEE-34 and BEE-35 would always refer to the same two bees, different bees might be *the-bee-I-am-chasing* at different times. Pengi uses its visual routines—patterns of directed visual activity—to register *aspects* of various entities, for example *the-bee-I-am-chasing-is-running-away*. The participatory, visually grounded nature of deictic representation means that Pengi is in constant interaction with its environment, rather than building and pondering models of it.

Let us consider a relatively complex example, starting from the situation illustrated schematically in Figure 1.

In this situation, the penguin (which is controlled by the Pengi system) wants to kill the enemy bee by kicking an ice cube at it. Ice cubes, when kicked, slide across the two-dimensional game board in a vertical or horizontal direction. Thus, to kill a bee,

an ice cube must be aligned with it in one of the two Cartesian dimensions. In this situation, no ice cube is aligned with the bee. However, if the penguin first goes over to the ice cube labeled *the-projectile-cube* and kicks it right, it will collide with the ice cube labeled *the-stop-cube* and come to a halt. (Energy is not conserved in this game.) *The-projectile-cube* will then be aligned with the bee, and can be kicked at it.

A planning system might approach this situation by constructing a four-step plan: go to left side of *the-projectile-cube*; kick *the-projectile-cube*; go to top of *the-projectile-cube*; kick *the-projectile-cube*. The executive that is given this plan must verify the plan's continued applicability by checking a long list of conditions that might have arisen to invalidate it: the bee might wander off, or another bee with hostile intentions might buzz in and need to be bumped off, or another bee might kick some ice cubes and thereby disturb the configuration in a way that makes carrying out the plan impossible.

Pengi constructs no plans and does no simulation. In place of simulation Pengi uses *visualization*. It engages in visual routines which find particular spatial configurations that predict courses of events and so suggest actions. For example, when Pengi sees that an ice cube adjacent to the penguin is aligned with a bee, and there are no intervening ice cubes, it kicks it, making it likely to strike and kill the bee. When it sees such an ice cube that is only near, rather than adjacent to, the penguin, it moves the penguin in the direction of the ice cube, because once it gets there the bee might still be aligned. If no ice cubes are aligned but the complex configuration of Figure 1 obtains, Pengi sends the penguin over to *the-projectile-cube* in order to kick it at *the-stop-cube*.

Put in the situation of Figure 1, Pengi may well engage in the same course of activity a planning system would, but for quite different reasons. Consider, for example, why each system would take the fourth and final action, kicking *the-projectile-cube* at the bee. The executive would take this action because the value of its program counter is four. Pengi takes the action because, by visualizing, it can see that by doing so it is likely to kill the bee. Once it has gotten to that point, it has no use for the idea that kicking that ice cube is part of a larger pattern of activity.

Even though Pengi's network is only partially implemented, it still plays a pretty decent game of Pengo. We started designing the network by envisioning a series of scenarios, which we call *routines*, of the common patterns of interaction between the player and the game. In practice, Pengi regularly exhibits these routines. What's more, Pengi regularly aborts a routine when a contingency arises, embarks on a new routine when an opportunity arises, interleaves different routines, and combines its repertoire of activities in useful ways we didn't anticipate. (It also regularly does silly things in situations for which we haven't yet wired it.)

Pengi illustrates some ideas, but Pengo-playing differs from other human activities in many ways. Most activities are less hectic, have more complex goal structures, require more remembering, involve additional kinds of representation such as visual imagery and

internal language, and so forth. Our experience with Pengi has focused the issues for a new round of study of dynamics and machinery.

Pengi, as we've mentioned, neither makes nor uses plans. Pengi engages in a continual, participatory interaction with its environment. Yet its activity is directed toward particular concrete goals: killing certain bees, staying clear of others, becoming adjacent to ice cubes it might usefully kick, and ultimately winning the game. Does this mean that plans are useless? Not at all. Pengi is a study of a certain subset of the dynamics of improvisatory activity. A creature that can participate in this set of dynamics can play Pengo.

Many other activities do require plans. For example, if Pengo got harder, Pengi might sometimes have to refer to a plan. The plan would explain how to deal with some tricky situation, or perhaps what strategic issues bear on the matter of which bees to attack when. The plan wouldn't be exhaustive like a program because Pengi isn't dumb like a programming language interpreter. Instead, the plan might consist of natural language, or something very much like it.

4 Plans as communication

The plan-as-program view and the plan-as-communication view differ as to the nature of plan use, the way in which plans are representations, and the nature of activity.

Nature of plan use. For the plan-as-program view, a plan decomposes into primitive actions which can be simply "emitted" by the executive, a simple, fixed, domain-independent device. For the plan-as-communication view, figuring out what activity a plan suggests requires a continual interpretive effort. It can take a lot of work to determine what in the situation the plan is talking about. A plan is operational if a sensible agent can use it, somehow, to engage in the activity it describes. A plan is a resource you can draw on in deciding what to do, on an equal basis with other resources such as the arrangement of your equipment, external memory devices like a string tied around your finger or a scratch pad, and your feelings. Unlike mechanical executives, people using plans know more or less what they are doing and why. Thus a plan is often well thought of as a mnemonic device.

Nature of representation. A plan-as-program "represents" a course of action in a very simple sense, insofar as programming languages have roughly compositional semantics. Each primitive of a programming language always occasions the same action, independent of "context." For the plan-as-communication view, a plan "represents" a course of action in a much more complex sense, insofar as a linguistic entity's meaning depends on the context of its use in a hundred different ways. In particular, a program represents its actions "exhaustively" where a linguistic entity cannot and need not.

On the plan-as-program view, plans are abstract mathematical entities. On the plan-

as-communication view, plans are social constructions (Hutchins 1987, Wertsch 1985). Children learn collaboration before they make plans for themselves. Our ability to make and use plans is built on our ability to use language during activities we share with others.

Nature of activity. In the plan-as-program view, the only situation given thorough consideration is the "initial situation" given to the planner. During the course of execution, the circumstances that arise can only determine conditional branches or cause control to be returned to the planner if something goes obviously wrong. The plan-as-communication view is part of a theory of "situated activity" (*cf.* Suchman 1987). Situated activity isn't some special variety of activity. The phrase emphasizes that a central feature of *all* activity is that it takes place in some specific, ongoing situation.

The plan-as-communication view suggests that the world's independence of your control is not an obstacle to be overcome but a resource to be exploited (*cf.* Suchman 1986). If your activity is not rigidly controlled by a plan, contingencies need not be disruptive; instead they can occasion creative improvisation.

In choosing the plan-as-communication view over the plan-as-program view, we implicitly promise to explain the role of plans-as-communications in a broader theory of situated activity. This is a big project. The remainder of this essay sketches some starting points.

Let's consider a typical example of human plan use. The route from my (Agre's) flat in Boston to the subway station, a distance of about three blocks, is hard to describe without drawing maps. (See Figure 2.) Nonetheless, we found that three experimental subjects unfamiliar with the area had no difficulty traversing the route using as a plan only "left out the door, down to the end of the street, cross straight over Essex then left up the hill, take the first right and it'll be on your left," which is nothing next to the actual complexity of the trip.

(This example might be disorienting in that the plan's maker and user are different people. We'll suggest later that using a plan you've made yourself is much like being instructed by someone else.)

Consider how much these directions leave out. "The door" is presumably the front door of the building. There's no need to tell you to walk down the street in the direction that "left out the door" will leave you headed; when you're on a path you don't need a plan. No need to label "down to the end" a figure of speech rather than an instruction to descend somewhere. No mention, either, of the fact that Essex Street is not marked as such anywhere near its intersection with Edinboro Street. There's no need to mention it, since it'll be clear which street is meant once you get there. (Our subjects reported being bothered by the lack of a sign but all of them proceeded correctly anyway.) "Left up the hill" will manage to refer to the Avenue de Lafayette rather than to Essex Street

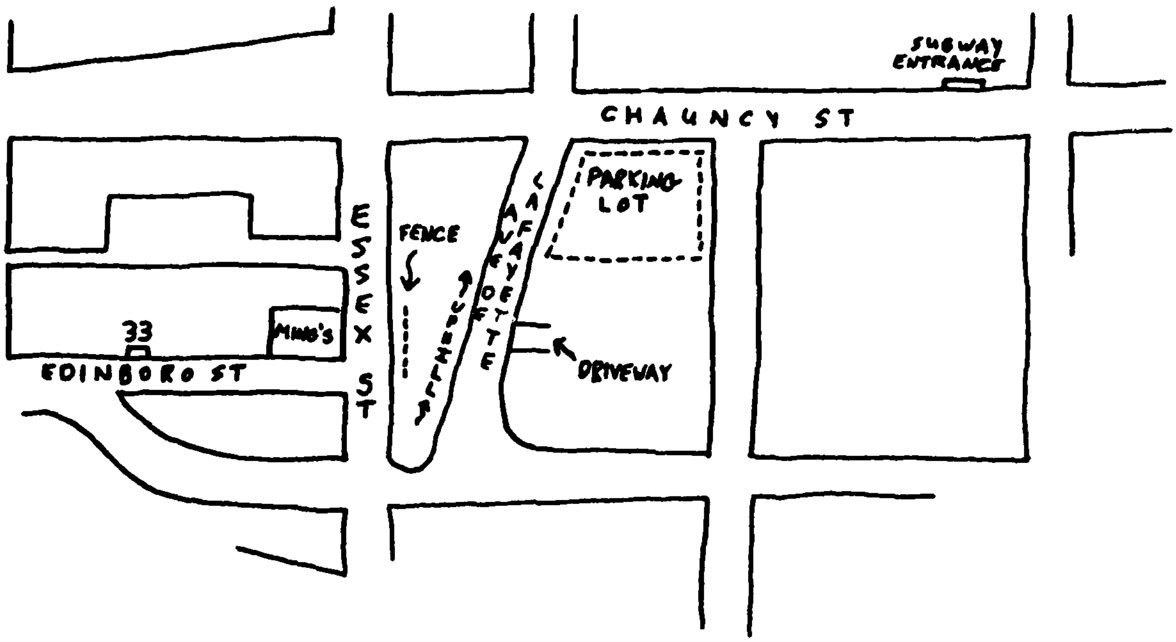

Figure 2. The route from 33 Edinboro Street to the Washington Street subway station, early 1986. (Not to scale.)

because it's the only hill you can see when you're standing at that intersection looking that way. Getting to the Avenue will require a brief rightward detour to get around a fence. No need to mention either this detour or the necessity of crossing the Avenue. When I walk this route myself I typically cut through a parking lot that precedes the "first right." The directions leave out the parking lot altogether; presumably you will have the sense to see the first right coming and cut the corner; and it doesn't matter if you don't. You'll also need the sense not to interpret a driveway or the parking lot itself as that first right. Everyone relies heavily on these sorts of things, usually without specifically knowing it, when giving directions. Some people are better at it than others. For example, experienced urban direction-givers know that alleys can confuse people who've been directed to count lefts or rights.

When you're using a plan, your surroundings are available as a resource for interpreting it. A plan that refers to "the hill" counts (roughly speaking) on there only being one hill apparent to someone who has gotten that far in the plan. A plan that instructs you to "take the first right" counts on it being clear which street is indicated. "Counting" and "clarity" are defined reflexively, almost circularly, as that which a given person will be able to figure out in a given situation. Just so you can figure it out once you get there.

The plan also relies on your experience and skill. The instruction to "walk down to the end of the street" assumes you have the sense to disobey it when the street is full of

slush or garbage or dangerous-looking people, as it often is. The plan omits things you already know, like how to cross a street, how to use street signs, how to detect another street coming up, and where it's safe and legal to walk. It also omits things you can be trusted to figure out for yourself, like how to recognize the subway station, how to wind your way past the trash strewn outside Ming's grocery, and how to get some new directions if you get lost.

In short, this plan exploits a long list of ways in which its maker and its user share an understanding of the world. We would like to suggest that this lesson generalizes in several ways: that the list of shared understandings is actually innumerably long; that all plans depend on shared understandings in this way; that action in the real world is sufficiently difficult to specify that plans *must* depend on innumerable shared understandings to be expressible at all; and that all of these points apply regardless of whether the plan's maker and user are the same agent or different agents. If true, these assertions appear to cast doubt on the possibility of a general plan-construction faculty. It follows that the skill of constructing useful plans must operate in some other fashion. Our hypothesis is that the human ability to make plans derives from our formative experiences with using language to communicate about ongoing situated cooperative activity. Our current work explores this view by starting with some simple but, we believe, representative cases.

5 Our current work

Reducing plan use to natural language comprehension might not sound very helpful. We certainly don't want to trivialize the role that natural language plays in situated activity; it's a big topic (see for example Heritage 1984, Stucky 1987). To get computational investigation started, we need to pick some simple, prototypical cases. We have started by trying to understand the role of language, and of communicative activity generally, in routine cooperative activities. The things we've learned apply to plan use. Plan use is a more complicated case of situated language use, if only because plans tend to be longer and more syntactically complicated than the utterances exchanged by participants in the course of an ongoing activity.

In order to understand the detailed connections between language use and physical activity, we study videotapes of people collaborating. For example, we have taped players of cooperative arcade games, in which two players, moving through a simulated maze, work together to fight off monsters. The players in these tapes are already good at video games and at the coordination required for cooperative play; in many cases they are expert at the particular game they are playing. As a result, their activity is largely routine. Moreover, the players see the same screen and have much the same understanding of the game, so they can depend on their shared understanding to achieve most coordination. Thus they need say very little. With rare exceptions, their talk serves only to repair

minute differences in understanding. One player might simply say "No!" because there are only two activities the other might plausibly undertake in the current situation. The utterance exploits their commonality of understanding to interpret the listener's moves as constituting a certain activity, judge that activity to be the wrong one, and suggest that he desist from that activity and instead join the speaker in the other one.

To take another example, very often on our tapes one player will say to the other "Turn left!" Most often, the other player does not immediately turn left. Yet this is not an error, nor is the advice erroneous, nor does the speaker consider that she has been disobeyed. In fact, a viewer will generally agree that the instruction was carried out. Activity other than immediately turning left can count as fulfilling the instruction in many domain-specific ways.

- In some cases, the doorway through which it will be possible to turn has not yet been reached, so that turning left would run you into a wall. In these cases, turning left is deferred.
- When the point at which a turn is possible is reached, there may also be a doorway on the right, and there may be an monster hiding behind the door. If the monster will shoot her in the back when she turns left, the player will turn right and kill the monster before turning back around and proceeding.
- In one case in our collection, the player passes the turn to pick up a valuable energy pod and then returns to comply with the instruction.
- Again, it may be that there is no left turn available, but there is an obviously correct right turn; in this case, the player may well figure that her interlocutor has simply said "left" for "right" in the heat of the moment, and turn right without comment.

The player is only likely to say "huh?" when she can make no sense at all of the instruction.

Not only can instructions be deferred; often they can be enacted with actions that, taken literally, violate them. For example, during a game of Gauntlet one player said "Don't go below that line," pointing at an imaginary line on the screen. Monsters in Gauntlet always head straight for you. Thus it is often important not to pass below the edge of a wall; if you do, monsters will stream around the corner and attack you. However, everyone eventually *did* go below that line without the instruction being explicitly rescinded; they mutually understood that it was now time to go after that particular set of monsters.

The players' utterances could be so compact because their possible import was heavily constrained by indexicality, projection, and reflexivity.

- *Indexicality.* We interpret communications with regard to the present circumstances. "No" offers advice about some ongoing activity whose manifestations

are visible to both players through the motions of one of the figures on the screen. "Turn left" picks out a certain corridor in the maze, one which is specified in terms of the listener's current location and heading. "Don't go below that line" picks out a certain *imaginary* line that the speaker can point at because both parties know to visualize it. In each case, the players are not making reference to objectively available "features" of the video screen but to shared interpretations of the commonly-visible whirl of colored lights.

- *Projection.* Each of us knows what might be expected to happen next. An imperative like "No," "Turn left," or "Don't go below that line" will typically invoke a projection of the specified course of events and another projection of the "or else" that might result if the listener disobeys. Skilled players will generally be able to perform both projections since they are familiar with the ways of the game.
- *Reflexiveness.* Each player understands that the players share an understanding of the situation; since the other player's understanding is part of the situation, this applies recursively. Player A can only expect "No" to communicate if player B understands herself to be engaged in the particular activity "No" recommends against; player B can only make sense of the instruction if she imagines that player A considers her to be engaged in that activity; player A must further be able to count on player B imagining this; and so on. Likewise, both players must reflexively understand that "Turn left" picks out a certain corridor and that "Don't go below that line" picks out a certain imaginary line.

To an amazing extent, the players assume that they both see the evolving game the same way, despite its large number of continually shifting issues. The players *must* make this assumption. If they didn't then they could never finish specifying everything that would be necessary to relate their advice to the evolving game situation. Indeed, we doubt if the players could list their shared understandings if they had to. Communication doesn't pick up a "meaning" from my head and set it down in yours. Instead, communication is part of the work of maintaining a common reality. The players shared a common reality because they were both competent players and because they used language to keep their shared reality in good repair.

6 Conclusion

We have outlined and contrasted two views of the nature of plans and plan use, the plan-as-program view and the plan-as-communication view. We have offered some reasons to doubt the plan-as-program view and speculated briefly about the nature of plans viewed as communications about situated activity. Specifically, we made three proposals:

1. The ability to make and use plans arises from, and is continuous with, one's experience with cooperative language use in the context of ongoing concrete activity.

2. In this regard, using one's own plans is much like using plans communicated by another.

3. Plan use, as a species of situated language understanding, relies on innumerable assumptions that the participants hold in common about the world and about the evolving concrete situation.

Many of the technical questions raised by the plan-as-communication view are as yet ill-defined, and certainly unanswered. Our initial ideas are only starting points. We do suggest, however, that research into the dynamics of plan making and plan use requires a worked-out view of the nature of everyday activity. Finally, we suggest that a critical and never-ending prerequisite to such an understanding is continual, detailed, sociologically informed observation of the ordinary everyday situated activity of the only truly successful plan makers we know of, namely human beings.

Acknowledgements

In working out these ideas about plans and planning, we were greatly aided by conversations with and comments on drafts by John Batali, Rod Brooks, Gary Drescher, Barbara Grosz, Beth Preston, Jeff Shrager, Orca Starbuck, and Ramin Zabih. Our thinking has been greatly influenced by Lucy Suchman's introducing us to ethnomethodology (Garfinkel 1967, Heritage 1984). The essay itself descends from two position papers we prepared for the DARPA Planning Workshop in Santa Cruz, California in October 1987. Thanks to Ted Linden and Drew McDermott for organizing the panels where they were presented and to Tomás Lozano-Pérez and Drew McDermott for helpful comments in Santa Cruz. Thanks also to Mike Brady, Rod Brooks, and Stan Rosenschein for various sorts of support.

References

Philip E. Agre, The dynamic structure of everyday life, MIT Computer Science PhD Thesis, forthcoming.

Philip E. Agre and David Chapman, Pengi: An implementation of a theory of activity, Proceedings of AAAI-87.

Philip E. Agre and David Chapman, Indexicality and the binding problem, Proceedings of the AAAI Symposium on Parallel Models, 1988.

King Beach, Becoming a bartender: The role of external memory cues at work, unpublished MS, Laboratory for Cognitive Studies of Work, CUNY Graduate Center, 1986.

David Chapman, Planning for conjunctive goals, *Artificial Intelligence*, **32** (1987) pp. 333-377.

David Chapman and Philip E. Agre 1986, Abstract reasoning as emergent from concrete activity, in M. P. Georgeff and A. L. Lansky (editors), *Reasoning about Actions and Plans*, Proceedings of the 1986 Workshop at Timberline, Oregon, pages 411–424, Morgan Kaufmann, Los Altos CA (1987).

R. T. Chien and S. Weissman, Planning and execution in incompletely specified environments, *Advance Papers of the Fourth International Joint Conference on Artificial Intelligence*, 1975, pages 169-174.

R. James Firby, An investigation into reactive planning in complex domains, Proceedings of AAAI-87.

Mark S. Fox and Stephen Smith, The role of intelligent reactive processing in production management, in *13th Meeting and Technical Conference*, CAM-I, November 1984.

Harold Garfinkel, *Studies in Ethnomethodology*, Polity Press, Oxford, 1984. Originally published in 1967.

Michael Georgeff and Amy Lansky, Procedural knowledge, Proceedings of the IEEE, Special Issue on Knowledge Representation, pp 1383-1398, October 1986.

Michael Georgeff and Amy Lansky, Reactive reasoning and planning, Proceedings of AAAI-87, pp 677-682.

Georges Giralt, Raja Chatila, and Marc Vaisset, An integrated navigation and motion control system for autonomous multisensory mobile robots, Proceedings of the First Symposium on Robotics Research, MIT Press, 1984, pp 191-214.

Thomas Gladwin, *East is a Big Bird*, Harvard University Press, 1970.

Edwin Hutchins, Learning to navigate in context. Manuscript prepared for the Workshop on Context, Cognition, and Activity, Stenengsund, Sweden, August 6–9, 1987. Institute for Cognitive Science, University of California, San Diego, La Jolla, California.

John Heritage, *Garfinkel and Ethnomethodology*, Polity Press, Cambridge, England, 1984.

Tomás Lozano-Pérez and Rodney A. Brooks, An approach to automatic robot programming, AI Memo 842, MIT Artificial Intelligence Laboratory, 1985.

Drew McDermott, Planning and Acting, *Cognitive Science*, **2**, 71–109, 1978.

George A. Miller, Eugene Galanter, and Karl H. Pribram, *Plans and the Structure of Behavior*, Henry Holt and Company, 1960.

Bob Scher, *The Fear of Cooking*, Houghton-Mifflin, 1984.

Sylvia Scribner, Studying working intelligence, in B. Rogoff and J. Lave, eds, *Everyday Cognition: Its Development in Social Context*, Harvard University Press, 1984.

Susan U. Stucky, The situated processing of situated language, CSLI Report 87-80, March 1987.

Lucy Suchman, What is a plan?, ISL Technical Note, Xerox Palo Alto Research Center, 1986.

Lucy Suchman, *Plans and Situated Action*, Cambridge University Press, 1987.

William Swartout, ed, DARPA Santa Cruz Workshop on Planning, *AI Magazine*, Summer 1988, pages 115-131.

Austin Tate, Planning and Condition Monitoring in a FMS, International Conference on Flexible Manufacturing Systems, London, UK, July 1984.

Austin Tate, A review of knowledge-based planning techniques, *The Knowledge Engineering Review*, volume 1 number 3, June 1985, pages 4-17.

Shimon Ullman, Visual Routines, MIT AI Memo 723, June, 1983.

James W. Wertsch, *Vygotsky and the Social Formation of Mind*, Harvard University Press, Cambridge MA, 1985.

Robert Wilensky, *Planning and Understanding: A Computational Approach to Human Reasoning*, Addison-Wesley, Reading MA, 1983.

David E. Wilkins, Recovering from execution errors in SIPE, SRI Tech Report 346, 1985.

David E. Wilkins, *Practical Planning: Extending the Classical AI Planning Paradigm*, Morgan Kaufmann Publishers, Los Altos CA, 1988.